Reducing the Magnitude of Transients Generated by the Model Reference Adaptive Control Method

Alexandru Magdalinoiu
Department of Electrical Engineering
Eindhoven University of Technology
Eindhoven, Netherlands
Student ID 1319086
a.magdalinoiu@student.tue.nl

***Abstract*—Model reference adaptive control (MRAC) is a variable parameter PID tuning method commonly used to combat plant nonlinearities. The adaptation process of the parameters, however, can prove to be either slow or lead to strong transient behaviour that is dangerous for a real–time implementation. Within this paper two control techniques are proposed for countering this problem: (i) a neural network based PID and (ii) a novel, hybrid controller which combines a fixed parameter internal model control PID with the considered MRAC tuned PID. Tests are carried on a simulation of an industrial coreless linear motor in order to assess their performance under both linear and nonlinear disturbances. The obtained results are discussed, and potential future research and improvements are suggested.**

I. Introduction

THE proportional–integral–derivative (PID) controllers with fixed parameters are currently used in most industrial applications due to their ease of implementation [1], [2]. While they achieve good performance when applied to linear plants, more complex systems operate under highly nonlinear conditions. As such, fixed parameter PID controllers can not ensure the precision required to achieve decent error rejection when stabilizing said systems. In order to combat problems such as nonlinear plants or changing system characteristics over time, model reference adaptive control (MRAC) tuning is used in an increasing number of applications such as first order systems [3]. These methods, however, can lead to powerful transient behaviour which may damage the plant or lead to very large errors at times.

This paper proposes two adaptive controllers that achieve similarly low errors as the MRAC PID, but without suffering from the harmful oscillations. Research is conducted in order to assess whether neural network PID (NPID) control can help alleviate the transient behaviour and limit the error by learning the required PID parameters. The second considered controller is a hybrid which combines MRAC with traditional fixed variable Skogestad internal model control (SIMC) [4]. The selected plant model used throughout the research is a coreless linear motor (CLM) used as part of the operation of a real—life lithography machine [5]. Performance is compared in simulation considering the nonlinear disturbances, while MRAC tuning using the MIT adaptation rule is considered the baseline performance. Section II details the operation and modelling of a CLM and presents the proposed control techniques. Section III reports on the experimental results, while Sections IV and V discuss the obtained insights, limitations of the experiments as well as possible points of improvement.

II. Methods

A. Operation and modelling of the CLM

The CLM is used in a wide range of applications due to its high precision [6], [7]. It can be modelled as two different parts, the electromagnetic and mechanical parts [5]. For the purpose of the performance test, the model assumes a commutation block that perfectly inverts the electromagnetic part, so only the mechanical motion of the CLM is considered.

The mechanical motion is represented by

$$\ddot{y}(t) = \frac{u(t) - F_{friction}(y(t), \dot{y}(t))}{m}, \quad (1)$$

where $y(t)$, $\dot{y}(t)$ and $\ddot{y}(t)$ are the position, velocity and acceleration of the CLM, $u(t)$ is the input force, $F_{friction}(y(t), \dot{y}(t))$ is the friction force and m is the mass.

We consider two types of friction. First, a linear friction model that only assumes the viscous friction which can be expressed as

$$F_{friction}(\dot{y}(t)) = f_v \dot{y}(t), \quad (2)$$

where f_v represents the viscous friction coefficient. Substituting (2) into (1) leads to the CLM transfer function

$$G = \frac{1}{ms^2 + f_v s}. \quad (3)$$

Secondly, a nonlinear model which was identified on the plant [8] is described by

$$F_{friction}(y(t), \dot{y}(t)) = f_c \text{sgn}(\dot{y}(t)) + f_v \dot{y}(t) - c_1 \sin(\omega y(t)) + (f_s - f_c)\text{sgn}(\dot{y}(t)) e^{-(\dot{y}(t)/v_s)^2}, \quad (4)$$

f_c, c_1 and f_s being the Coulomb, sinusoidal and Stribeck friction coefficients, and v_s is the Stribeck velocity coefficient [5].

It should be noted that the linear model is used for controller tuning, whereas the performance was evaluated using the nonlinear model. The values of the aforementioned parameters are listed in Table I and the implementation of the nonlinear friction plant is shown in Fig. 1.

TABLE I
PRACTICAL COEFFICIENT VALUES OF THE CLM

Parameter	Variable	Value	Unit
m	Mass	20	kg
f_c	Coulomb coefficient	8.18	N
f_v	Viscous coefficient	135.75	Ns/m
c_1	Sinusoidal coefficient	1.44	N
ω	Period of the position dependency	2.21	rad/m
f_s	Stribeck coefficient	14.63	N
v_s	Stribeck velocity	0.0123	m/s

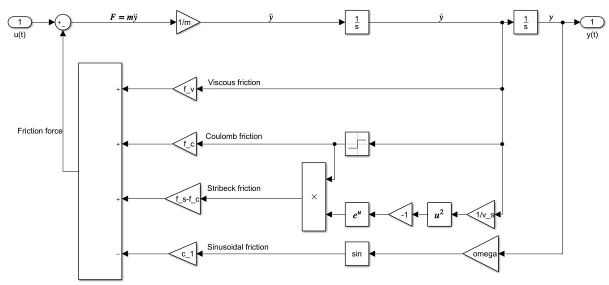

Fig. 1. The implementation of the nonlinear friction plant.

B. Skogestad internal model control tuning

The preferred PID architecture in the current paper is the parallel implementation, as it is the most intuitive for understanding the influence of each parameter. This can be expressed in the time domain as

$$u(t) = K_P e(t) + K_I \int_0^t e(t) d\tau + K_D \frac{de(t)}{dt}, \qquad (5)$$

where $e(t) = r(t) - y(t)$ is the error between the desired (r) and actual (y) system output. The controller is placed in closed-loop with the plant described by (3) in order to stabilize it.

Skogestad derived tuning rules for an integrating delay process with lag τ_2, which can be modelled as

$$g(s) = k' \frac{e^{-\theta s}}{s(\tau_2 s + 1)}, \qquad (6)$$

where k' is the plant gain. Due to the displayed dynamics, (3) can be regarded as a special case of (6). In order to obtain the tuning rules, a desired closed–loop first–order response

$$\frac{y}{r} = \frac{e^{-\theta s}}{\tau_c s + 1}, \qquad (7)$$

is specified in [4], where θ is the time delay and τ_c is the tuning parameter. By relating (3) and (6), the values of the parameters are found: $k' = 135.8$, $\tau_2 = \frac{m}{f_v} = 0.1473$ and $\theta = 0$. In reality, the considered CLM processes inputs with a sampling rate of 10kHz, and as such a value of $\theta = 10^{-4}$ can be considered.

Generally, the time constant τ_c needs to be larger than θ in order to ensure the system is robust, however in the case of the considered plant such a value is unrealistically small. For the purpose of this paper, a value $\tau_c = 0.03$ was selected as it leads to a fast response while keeping the input force magnitude within reasonable margins. The values suggested in [4] for tuning a Cascade PID are

TABLE II
CONTROL PARAMETERS COMPUTED BY SKOGESTAD'S METHOD

Parameter	Value
τ_c	0.03
K_P	10080.55
K_I	37708.33
K_D	666.67

$$K_P = \frac{1}{k'} \frac{1}{\tau_c + \theta} \Big(1 + \frac{\tau_2}{4(\tau_c + \theta)}\Big), \qquad (8)$$

$$K_I = \frac{1}{4k'} \frac{1}{(\tau_c + \theta)^2} \text{ and} \qquad (9)$$

$$K_D = \frac{1}{k'} \frac{1}{\tau_c + \theta} \tau_2. \qquad (10)$$

The values of the control parameters are summarized in Table II. It should be noted that selecting a different value of τ_c will dramatically influence the behaviour of the controller: a higher value leads to more robust and sluggish behaviour, while a lower value leads to a less stable tune but better overshoot and settling time characteristics.

C. Model reference adaptive control tune

This technique allows the PID parameters K_P, K_I and K_D to adapt in real time by minimizing the discrepancies between the plant output y and an a priori chosen reference model output y_m. The reference model is the same as the desired closed–loop first–order response used by Skogestad for the IMC tuning, shown in (7).

The closed–loop response of the system can be computed by expressing the parallel PID shown in (5) in Laplace domain and plugging in the plant transfer function from (3), leading to

$$\frac{Y(s)}{R(s)} = \frac{\frac{1}{m}(K_D s^2 + K_P s + K_I)}{s^3 + (\frac{f_v}{m} + \frac{1}{m}K_D)s^2 + \frac{1}{m}K_P s + \frac{1}{m}K_I}. \qquad (11)$$

[3] proposes the MIT rule for value updating which minimizes the cost function

$$J = \frac{\epsilon^2}{2}, \qquad (12)$$

where $\epsilon = y(t) - y_m(t)$. As the optimization problem uses a gradient descent approach, the rate of change of the PID parameters can be expressed as

$$\frac{dK_\alpha}{dt} = -\gamma_\alpha \frac{\partial J}{\partial K_\alpha} = -\gamma_\alpha \frac{\partial J}{\partial \epsilon} \frac{\partial \epsilon}{\partial y} \frac{\partial y}{\partial K_\alpha}, \ \alpha \in \{P, \ I, \ D\}, \qquad (13)$$

γ_α being the adaptive gain for each of the parameters. By substituting (12) and (11) in (13) the final update laws can be computed as

$$\frac{dK_P}{dt} = -\gamma_p \epsilon \frac{\frac{1}{m} e s}{s^3 + (\frac{f_v}{m} + \frac{1}{m}K_D)s^2 + \frac{1}{m}K_P s + \frac{1}{m}K_I} \qquad (14)$$

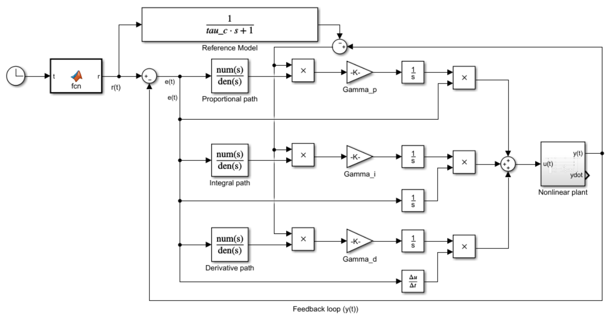

Fig. 2. The used MRAC PID applied on the nonlinear CLM model.

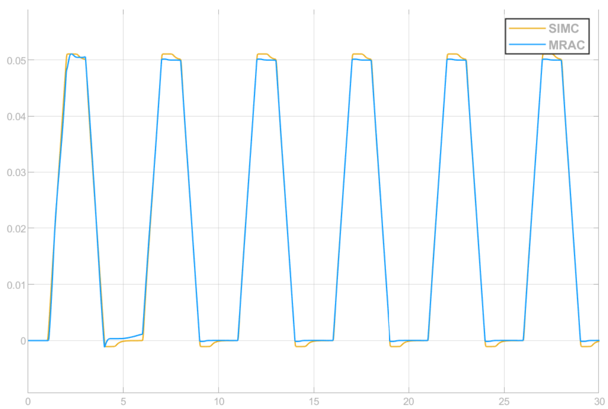

Fig. 3. Comparison between the SIMC and MRAC performance.

$$\frac{dK_I}{dt} = -\gamma_i \epsilon \frac{\frac{1}{m}e}{s^3 + (\frac{f_v}{m} + \frac{1}{m}K_D)s^2 + \frac{1}{m}K_P s + \frac{1}{m}K_I} \quad (15)$$

$$\frac{dK_D}{dt} = -\gamma_d \epsilon \frac{\frac{1}{m}es^2}{s^3 + (\frac{f_v}{m} + \frac{1}{m}K_D)s^2 + \frac{1}{m}K_P s + \frac{1}{m}K_I}. \quad (16)$$

An implementation of the MRAC PID controller can be observed in Fig. 2.

The advantage of this controller is that it manages to achieve better error rejection on the nonlinear plant compared to a fixed parameter controller, for example the aforementioned SIMC PID (as shown in Fig. 3). However, large errors may occur at the beginning of the process: transient behaviour bound to add huge errors or even damage the systems. This oscillating motion can be observed in the first 6.5 s of Fig. 3. The following controllers aim to diminish this behaviour while still achieving good error rejection.

D. Neural networks based PID

In the control field, neural networks can prove to be useful due to their ability to approximate functions of any order [9] given that enough neurons are present in the network. A multilayer perceptron (MLP) is used to approximate the relation between the current state of the system and the desired controller gains, similarly to the idea proposed in [10]. Whereas [10] does not indicate how to find these optimal gains, we exploit the use of the MRAC controller that achieves the desired gains after the adaptation process converges.

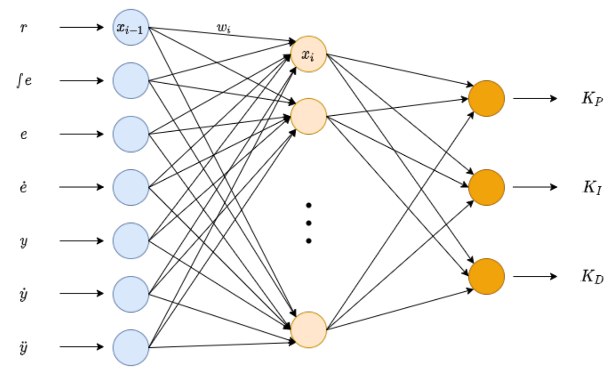

Fig. 4. The architecture of the neural network.

Starting from the plant dynamics (1) and the parallel PID controller (5) in order to select the relevant system states, the following input part of the dataset can be considered:

$$[r, \textstyle\int e, e, \dot{e}, y, \dot{y}, \ddot{y}]^T. \quad (17)$$

Given that the purpose of the NN is to reach the PID parameters, the target part of the dataset will naturally be

$$[K_P, K_I, K_D]^T, \quad (18)$$

such that the structure shown in Fig. 4 is obtained. The output of layer i is computed as

$$x_i = f(w_i x_{i-1} + b_i), \quad (19)$$

where w_i are the weights of the connections, b_i are the bias quantities and f is the sigmoid activation function

$$f(x) = \frac{1}{1+e^{-x}}. \quad (20)$$

The learning process consists of finding the optimal values for all weights and biases, which is achieved by using the Levenberg–Marquardt backpropagation algorithm [11] on a set of training data consisting of (17) and (18). This dataset can be obtained by measuring the states of the MRAC controlled plant. By only considering the period of time where no transient behaviour is encountered, the NPID is able to learn and mimic the good error–rejection achieved by MRAC while removing some of the oscillatory motion.

In order to generate persistently exciting data, a sinusoidal signal of amplitude 0.002 m and frequency 10π rad/s was added to the reference signal when measuring the relevant parameters.

The NPID controller research is not yet finished and results are expected in the following weeks.

E. Hybrid–α controller

Another method to reduce the undesired transients generated by the MRAC controller is by using a hybrid combination of the fixed and variable parameter PID controllers. The aim of

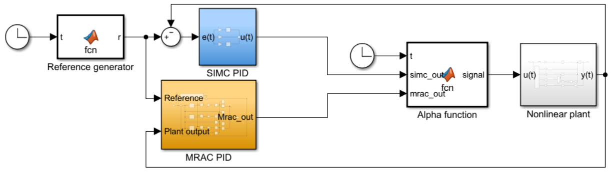

Fig. 5. Implementation of the *time* function hybrid controller.

this approach is to achieve the robustness of the SIMC tune and the performance of the MRAC tune.

The controller output can, for example, be defined as

$$u(t) = \alpha \cdot u_M(t) + (1-\alpha) \cdot u_S(t), \quad \alpha = [0,1], \tag{21}$$

where $u_M(t)$ is the MRAC control output, $u_S(t)$ is the SIMC control output and α is the quantity that splits the controller action between the fixed and adaptive controllers. The higher this value, the better error rejection the system will achieve, and one can easily observe that if α equals 1 throughout the experiment, the system will behave entirely as dictated by the MRAC PID. In contrast, an α value of 0 will lead to the fixed parameter PID controlling the plant.

The choice of α can depend on either time or instantaneous error compared to the reference signal.

- *Time* function – the value of α increases with time, in order for the fixed parameter PID to control the first part of the experiment, while allowing MRAC to control the plant after the transients disappeared. This rule ensures that the error is minimized as the plant is functional, and can be summarized as:

$$\alpha = \begin{cases} 0 & \text{, if } t \le \tau_1 \\ \dfrac{t-\tau_1}{\tau_2-\tau_1} & \text{, if } \tau_1 < t < \tau_2 \\ 1 & \text{, if } t \ge \tau_2 \end{cases} \tag{22}$$

  where τ_1 and τ_2 are the tuning parameters.
- Function of *error* – there are two relevant instantaneous error values that are used as tuning parameters: ε_1 and ε_2. By measuring the instantaneous error ε, it can be deduced if the MRAC controller is going through an oscillatory phase, so its influence can be decreased, as described by:

$$\alpha = \begin{cases} 0 & \text{, if } \varepsilon \ge \varepsilon_2 \\ \dfrac{\varepsilon_2-\varepsilon}{\varepsilon_2-\varepsilon_1} & \text{, if } \varepsilon_1 < \varepsilon < \varepsilon_2 \; . \\ 1 & \text{, if } \varepsilon \le \varepsilon_1 \end{cases} \tag{23}$$

An implementation of the α adaptation algorithm is shown in Fig. 5.

Simulation results of the Hybrid–α controller as a function of instantaneous error will follow in the upcoming weeks.

III. Performance evaluation

The Simulink software was used for the purpose of simulating the plant and controller interactions. In order to excite the system in a similar manner to the actual working conditions of this type of CLM, the reference signal was set to repeat the following motion:

TABLE III
RMSE Values of the tested controllers

Evaluated controller	Linear plant	Nonlinear plant
SIMC	$3.29*10^{-4}$ m	$5.75*10^{-4}$ m
MRAC	$7.97*10^{-4}$ m	$5.23*10^{-4}$ m
Hybrid-α	$5.61*10^{-4}$ m	$3.16*10^{-4}$ m

$$y = \begin{cases} 0, & \text{for } t \in [4n, 4n+1), \\ 0.05(t-4n), & \text{for } t \in [4n+1, 4n+2), \\ 0.05, & \text{for } t \in [4n+2, 4n+3), \\ 0.05 - 0.05(t-4n-3), & \text{for } t \in [4n+3, 4n+4) \end{cases}$$

where $n = 0, 1, 2, \dots$.

The assessment of the controllers' performance was done in terms of the root mean squared error (RMSE), as show in Table III.

A. Linear friction model

The main purpose of considering the linear friction model from (2) is to derive the SIMC and MRAC controller equations, but it can also be used to simulate an "ideal" environment in which the motor is going to run. As such, the NPID and Hybrid–α controllers should outperform the MRAC in these conditions as well as the nonideal scenario.

Fig. 6 shows the output of the three considered controllers. The transient behaviour of the MRAC PID is very visible throughout the first 10 s of the experiment, whereas the error that occurs after parameter adaptation is almost negligible.

Tuning the *time*–function Hybrid–α controller consists of selecting the values of τ_1 and τ_2 from (22). τ_1 dictates the time at which α begins to increase and τ_2 sets the rate of increase. After testing, the following insights were uncovered:

- τ_1 – the ramp motion has to start early to minimize SIMC action time to achieve lower average errors. However, the adaptation process needs 5 s to reach good Kp, Ki, Kd values before coupling. Moving the value of τ_1 below 5 leads to an increase in the transient response which has to be minimized: this is due to the SIMC and MRAC working "against" each other. As a consequence, the MRAC can not follow its normal adaptation process, and it is permanently hindered.
- τ_2 – as with τ_1, the value of this parameter should theoretically be as low as possible to minimize the action time of the fixed parameter controller. Making this value too small leads to a very abrupt increase in MRAC influence which can end up generating more transients.

The best results were achieved for values of $\tau_1 = 5s$ and $\tau_2 = 7s$, respectively.

Table III indicates that the Hybrid–α controller achieves a 42% lower RMSE value than the standard MRAC controller. As shown in Fig. 6, this is mainly attributed to the reduction in the occurring transients. Although the hybrid controller

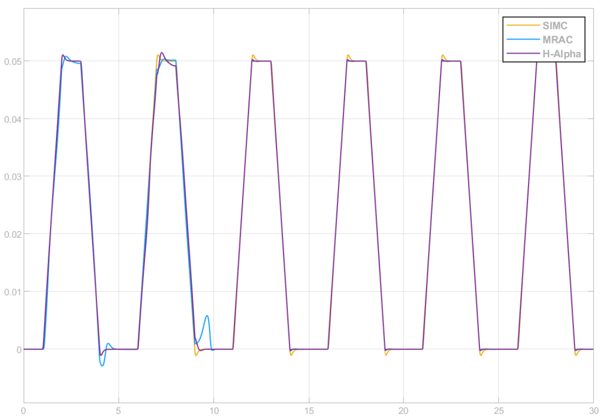

Fig. 6. The SIMC, MRAC and H–α controllers action on the linear plant.

achieves a more steady output, it can be noticed that the error after the initial 15 s are slightly higher than those made by the MRAC PID. This can be explained by the slower adaptation process: (14)-(16) show that the adaptation rate is influenced by the error $e = u - y$ among others. In the case of the MRAC, the occurring fluctuations lead to high errors which in turn help the parameters react faster. Whereas transients are bound to start immediately when the MRAC controls the plant, the Hybrid–α will only display transient behaviour once the MRAC is coupled. This means that, while it helps mitigating most of the oscillations, it has a lower rate of adaptation than the MRAC PID.

The proposed controller does not manage to achieve a better overall error than the fixed parameter PID, in spite of the fact that it achieves considerably lower instantaneous errors after the PID parameters have adjusted properly. A possible solution to this could be the implementation of an *error*–function hybrid controller, since it will allow the MRAC action to be decreased more dynamically than the current architecture. The *error*–function PID will be tested in the following weeks.

B. Nonlinear friction model

The behaviour of the three considered controllers can be observed in 7. While the fixed parameter PID experiences an increase of about 75% in its average error, the two adaptive controllers achieve lower overall errors in this case. This is a good showcase of the potential advantages of MRAC, as well as its unpredictable adaptation patterns.

Fig. 8 shows that the three MRAC parameters experience large jumps in the 0–6 s timeframe, in order to follow the initial increase in reference signal. They then begin steadily decreasing, which signals that the adaptation behaviour reached convergence and no transients will follow.

As in the case of the linear friction plant, the Hybrid–α controller achieves a lower RMSE than the standard MRAC controller, and, as expected, performs better than the fixed parameter controller. Furthermore, the proposed controller manages to decrease the amount of force that acts on the plant. In order to achieve such low overshoot and quick settling time, the MRAC controller currently outputs 320 N of force, and peaks at around 400 N. In contrast, the hybrid controller outputs a maximum of 140 N, as shown in Fig. 9.

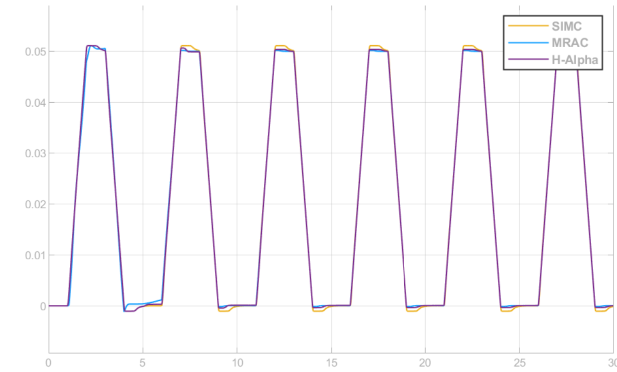

Fig. 7. The SIMC, MRAC and H–α controllers action on the nonlinear plant.

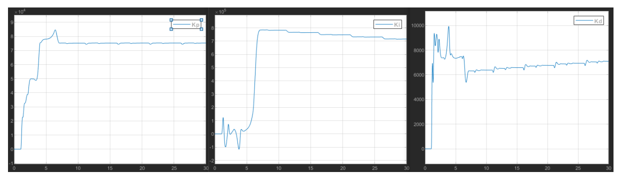

Fig. 8. The adaptation patterns of K_p, K_i, and K_d, respectively.

IV. Discussion

The hybrid controller managed to outperform the MRAC PID under both linear and nonlinear friction conditions. Whereas, for the purpose of the current study, the most visible difference is in terms of RMSE, the potential benefits of this controller are higher when used on a real CLM.

The most noticeable improvement achieved by the proposed setup is the fact that it is more predictable than the MRAC: by selecting τ_1 and τ_2, one can easily reduce the effect of the transient behaviour on the plant. This will possibly be improved further by the implementation of the *error*–function, as the *time*–function requires manual tuning of the parameters and is not too flexible in the adaptation process. An added benefit of reducing the force applied to the plant by more than 50% is the increased efficiency of the CLM, while the wear on the motor as consequence of functioning is also improved.

It should be noted that the proposed methods can help improve different adaptive controller types suffering from

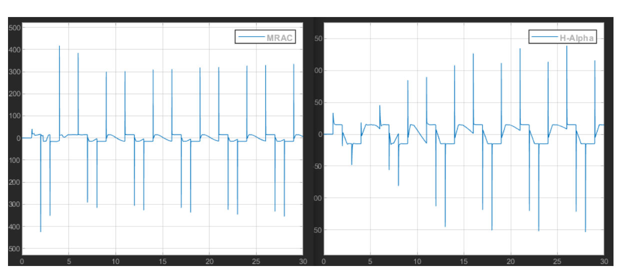

Fig. 9. Force output of the MRAC and Hybrid–α controllers.

occurrence of transients, and are not limited to MRAC tuning. The current results will be improved upon in the upcoming weeks as the NPID is implemented, alongside a more complex hybrid controller algorithm.

There are several improvements on which further work could be based on. Either considering adaptive neural networks or performing more in–depth testing regarding the neural network structure, such as testing other architectures than MLP. Using different activation functions, such as the Rectified Linear Unit, or algorithms other than Levenberg–Marquardt might improve the training process of the network and lead to more accurate K_p, K_i and K_d. Validating the obtained results on a real setup would be another possible advancement. Doing so could, for example, help define more clear guidelines for what τ_c values are realizable in practice on a CLM. Experimentation could also help quantify the efficiency improvement of the novel hybrid controller over the baseline MRAC PID.

V. Conclusion

In this paper two methods of improving the transient behaviour of an MRAC PID were analyzed. A fixed–parameter and an adaptive controller were tuned for establishing baseline performance, while a neural networks based PID controller and a novel hybrid PID were implemented with the purpose of improving the previous results. Tests were conducted on both a linear and a nonlinear friction model of a CLM and concluded that the hybrid controller manages to achieve (i) similar robustness to the SIMC tune and (ii) similar performance to the MRAC after parameter convergence. The neural networks PID will be tested in the following weeks.

How much this work actually improved the behaviour of the MIT rule MRAC depends on how the optimization problem is defined. If the process is largely error–tolerant but requires the quickest possible parameter adaptation, then MRAC tuning will be the better choice. However, if the plant is prone to being damaged by oscillation, large overshoot, or high input magnitudes, the proposed methods might be of interest. Dynamical processes will also benefit by the steadier adaptation patterns and increased process efficiency.

References

[1] K. H. Ang, G. Chong, and Y. Li, “PID control system analysis, design, and technology,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 7 2005.

[2] S. Wakitani, K. Nishida, M. Nakamoto, and T. Yamamoto, *Design of a data-driven PID controller using operating data*. IFAC, 2013, vol. 46, no. 11 PART. [Online]. Available: http://dx.doi.org/10.3182/20130703-3-FR-4038.00077

[3] B. Singh and V. Kumar, “A real time application of model reference adaptive PID controller for magnetic levitation system,” in *2015 IEEE Power, Communication and Information Technology Conference, PCITC 2015 - Proceedings*. IEEE, 2016, pp. 583–588.

[4] S. Skogestad, “Probably the best simple PID tuning rules in the world,” *researchgate.net*, 2001.

[5] Y. H. Yuen, M. Lazar, and H. Butler, “Data-driven neural feedforward controller design for industrial linear motors,” *2019 23rd International Conference on System Theory, Control and Computing, ICSTCC 2019 - Proceedings*, pp. 461–467, 2019.

[6] Q. Chen, “Real-time Model Predictive Control of Industrial Linear Motors,” Ph.D. dissertation, Eindhoven University of Technology.

[7] J. F. Gieras, Z. J. Piech, and B. Tomczuk, *Linear Synchronous Motors*, 2nd ed. New York, NY 10017: CRC Press, 2011.

[8] P. van den Boom, “Adaptive Friction Compensation for an Industrial Coreless Linear Motor Setup,” Ph.D. dissertation, Eindhoven University of Technology, 2018.

[9] M. T. Hagan, H. B. Demuth, and O. D. E. Jesús, “An Introduction to the use of Neural Networks in Control Systems,” *Endeavour*, vol. 25, no. 95, p. 58, 1966.

[10] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, “Neural network-based self-tuning PID control for underwater vehicles,” *Sensors (Switzerland)*, vol. 16, no. 9, pp. 1–18, 2016.

[11] J.J.More, “The Levenberg-Marquardt Algorithm Implementation and Theory,” *Numerical analysis*, no. x, pp. 9–12, 1978.